*J. Symbolic Computation* (1988) **5**, 275–283

# A Proof of Holt's Algorithm

GREGORY BUTLER

*Basser Department of Computer Science, University of Sydney, Sydney, NSW 2006, Australia*

*(Received 1 June 1986)*

Holt's algorithm obtains a p-step central series and a power commutator presentation for a p-group given by permutations that initially form a subnormal series. The permutations also contain a strong generating set for each group in the series. A proof of the algorithm, based on assertions, is presented.

## 1. Introduction

Several powerful algorithms have been developed by Felsch and Neubüser(1979) and Laue, Neubüser, and Schoenwaelder(1984) to investigate the structure of a p-group. These algorithms utilise an effective representation for a p-group, namely a power-commutator-presentation (pcp) based on a p-step central series. One common method of obtaining a p-group is as a Sylow subgroup of a permutation group. An algorithm of Butler and Cannon(1979) determines not only a base and strong generating set for the Sylow subgroup but also chooses a strong generating set that gives a subnormal series. The aim of Holt's algorithm is to obtain a p-step central series (and hence a pcp) for p-groups represented as a permutation group with such a base and strong generating set.

Holt(1984) developed the algorithm, but only described it briefly. The algorithm and a detailed proof are presented here. The corresponding implementation forms part of the Cayley system. The way we choose to look at the algorithm is as a conversion from a subnormal series to a p-step central series. Viewed in this way it is not too difficult to prove correct. We present this conversion process and its proof first. The complete algorithm must also preserve a strong generating set for each group in the series, in order for the steps of the conversion process to be effectively computable. This is done by noting the conditions under which the conversion process violates the strong generating set and in those circumstances replacing a generator by a more appropriate element of the same group. The groups in the series are not changed by the replacement, just the choice of generators. The proof that the strong generating sets are preserved is more difficult, so we treat it separately from the conversion process.

0747–1717/88/030275+09 $03.00/0

We assume the reader is familiar with both group theory and the theory of permutation groups as needed for computational purposes. The relevant material can be found in Butler(1982,1983), Butler and Cannon(1982), Hoffman(1982), Leon(1980), Macdonald(1968) and Wielandt(1964). The algorithms are presented using the control statements of Pascal with indentation indicating the range of a control statement. Comments are enclosed in braces { }. They indicate assertions about the values of variables at certain states of the execution of an algorithm.

## 2. Converting to a p-step Central Series

### 2.1 NOTATION

Our notation assumes the existence of a p-group G with a sequence of elements $g(1), g(2), \ldots, g(n)$ of G where $g(i) \notin < g(1), g(2), \cdots g(i-1) >$. We denote the group $< g(1), g(2) \cdots g(i) >$ by $G(i)$, so there is a chain of subgroups *identity* $< G(1) < G(2) \cdots < G(i)$.

There are several predicates that arise in the algorithms and their proofs. We list them here.

**SN($i$)**: the chain of subgroups *identity* $\lhd G(1) \lhd G(2) \cdots \lhd G(i)$ forms a subnormal series.

**p($i$)**: the chain of subgroups *identity* $< G(1) < G(2) \cdots < G(i)$ is p-step. That is, $| G(j) : G(j-1) | = p$, for all $j \leq i$.

**Z($i$)**: the chain of subgroups *identity* $\lhd G(1) \lhd G(2) \cdots \lhd G(i)$ is a central series. That is, $G(j) \lhd G(i)$ and $\frac{G(j+1)}{G(j)} \leq Z(\frac{G(i)}{G(j)})$, for all j < i.

**Z($i,j$)**: the chain of subgroups *identity* $\lhd G(1) \lhd G(2) \cdots \lhd G(i)$ is a subnormal series and $G(i)$ centralizes $G(k+1)$ modulo $G(k)$, for all $k < j$. That is, $[g,h] \in G(k)$, for all $g \in G(i)$ and $h \in G(k+1)$. Note that $j \leq i$.

Some consequences of these definitions are that **Z**($i$) implies **Z**($i,j$) for all $j \leq i$, and **Z**($i$) if and only if **Z**($i,i$). Since all the groups of order $p^2$ are abelian, p(2) implies **Z**(2), and **Z**($i,i-2$) implies **Z**($i$).

If **SN**($i$) and p($i$), then each element g of $G(i)$ can be uniquely written in the form $g = g(i)^{\varepsilon(i)} g(i-1)^{\varepsilon(i-1)} \cdots g(1)^{\varepsilon(1)}$, $0 \leq \varepsilon(j) < p$. This form is called the *normed word* of g.

During the execution of the algorithm, the sequence of elements $g(1), g(2), \cdots$ changes. If we wish to emphasis a value of the sequence other than the current value, we will use a bar, viz $\bar{g}(1), \bar{g}(2), \cdots$, on the sequence, groups, and predicates.

### 2.2 ALGORITHM

Given a sequence of elements $\underline{\bar{g}}(1), \underline{\bar{g}}(2), \ldots, \underline{\bar{g}}(n)$ that form a subnormal series of a p-group G of order $p^m$, *identity* $\lhd \bar{G}(1) \lhd \bar{G}(2) \cdots \lhd \bar{G}(n) = G$ return a sequence of elements

$g(1), g(2), \ldots, g(m)$ that form a p-step central series of G.

The algorithm obtains a p-step subnormal series by inserting powers of $\bar{g}(i)$ between $\bar{g}(i-1)$ and $\bar{g}(i)$, if necessary. To obtain a central series, the algorithm loops over i and j verifying $\mathbf{Z}(i,j)$. By induction, we can assume $\mathbf{Z}(i-1)$ and $\mathbf{Z}(i,j-1)$. The missing piece of information needed to conclude $\mathbf{Z}(i,j)$ is whether $g(i)$ centralizes $g(j)$ modulo $G(j-1)$. That is, whether the commutator $[g(j),g(i)]$ is in $G(j-1)$. If the commutator is not, then we insert the commutator between $g(j-1)$ and $g(j)$, deleting the now redundant generator $g(k)$, for some k. Eventually, some commutator $[g(j),g(i),g(i),g(i),\cdots]$ will lie in $G(j-1)$ thus verifying $\mathbf{Z}(i,j)$.

Recall that groups of order $p^2$ are abelian, so that $g(i)$ always centralizes $g(i-1)$ modulo $G(i-2)$ in a p-step subnormal series.

**Algorithm 1 : Form a p-step central series**

Input : a sequence of elements $g(1),g(2),\ldots,g(n)$ forming
a subnormal series of a p-group $G$;
Output : a sequence of elements $g(1),g(2),\ldots,g(m)$ forming a
p-step central series of $G$;
**begin**
$\{\mathbf{SN}(n)\}$
add in powers of elements to obtain a p-step series;
$\{\mathbf{SN}(m) \wedge \mathbf{p}(m) \wedge \mathbf{Z}(2)\}$
**for** $i := 3$ **to** $m$ **do**
$\{\mathbf{SN}(m) \wedge \mathbf{p}(m) \wedge \mathbf{Z}(i-1)\}$
**for** $j := 1$ **to** $i$-2 **do**
$\{\mathbf{SN}(m) \wedge \mathbf{p}(m) \wedge \mathbf{Z}(i-1) \wedge \mathbf{Z}(i,j-1)\}$
**while not** $[g(j),g(i)] \in G(j-1)$ **do**
$\{[g(j),g(i)] \notin G(j-1) \wedge [g(j),g(i)]^p \in G(j-1) \wedge \mathbf{SN}(m) \wedge \mathbf{p}(m) \wedge \mathbf{Z}(i-1) \wedge \mathbf{Z}(i,j-1)\}$
insert( $[g(j),g(i)], j$ );
$\{\mathbf{SN}(m) \wedge \mathbf{p}(m) \wedge \mathbf{Z}(i-1) \wedge \mathbf{Z}(i,j-1)\}$
$\{\mathbf{SN}(m) \wedge \mathbf{p}(m) \wedge \mathbf{Z}(i-1) \wedge \mathbf{Z}(i,j)\}$
$\{\mathbf{SN}(m) \wedge \mathbf{p}(m) \wedge \mathbf{Z}(i)\}$
$\{\mathbf{Z}(m) \wedge \mathbf{p}(m)\}$
**end.**

**procedure** insert( $g = g(i-1)^{\varepsilon(i-1)} g(i-2)^{\varepsilon(i-2)} \cdots g(1)^{\varepsilon(1)}, j$ );
**begin**
$\{g \in G(i-1) \wedge g \notin G(j-1) \wedge g^p \in G(j-1) \wedge \mathbf{SN}(m) \wedge \mathbf{p}(m) \wedge \mathbf{Z}(i-1) \wedge \mathbf{Z}(i,j-1)\}$
let $k$ be the highest integer such that $\varepsilon(k) \neq 0$;
$\{k \geq j\}$
delete $g(k)$; insert $g$ between $g(j-1)$ and $g(j)$;
$\{\mathbf{SN}(m) \wedge \mathbf{p}(m) \wedge \mathbf{Z}(i-1) \wedge \mathbf{Z}(i,j-1)\}$
**end;**

## 2.3 PROOF OF PROGRAM CORRECTNESS

The difficult part of the proof is verifying the pre- and post-conditions for the procedure insert, and the termination of the while loop. We begin by stating a well-known fact.

Lemma 1 : $\mathbf{SN}(i) \wedge \mathbf{p}(i)$ implies for all $g \in G(i)$ there is a unique expression $g = g(i)^{\varepsilon(i)} g(i-1)^{\varepsilon(i-1)} \cdots g(1)^{\varepsilon(1)}$, $0 \leq \varepsilon(j) < p$. []

Lemma 2 : The pre-conditions of insert are satisfied when it is called.

Proof

The conditions not involving the argument $g$ are obviously satisfied. Let $g = [g(j), g(i)]$ be the argument for the call to insert. Since $G(i-1)$ is normal in $G(i)$, and $g(j) \in G(i-1)$, then the commutator is in $G(i-1)$. Hence, $g \in G(i-1)$. Clearly $g \notin G(j-1)$. Let $h = g(j)^{g(i)}$. Then $g = g(j)^{-1}h$ and $h \in G(i-1)$. The commutator $[h, g(j)] \in G(j-1)$, by $\mathbf{Z}(i-1)$. Hence,

$$\begin{aligned} g^p &= (g(j)^{-1}h)^p = g(j)^{-p}h^p \text{ modulo } G(j-1). \\ &= h^p \text{ modulo } G(j-1), \text{ by } \mathbf{p}(m) \\ &= (g(j)^p)^{g(i)} \text{ modulo } G(j-1) \\ &\in G(j-1)^{g(i)}, \text{ by } \mathbf{p}(m) \\ &= G(j-1), \text{ by } \mathbf{Z}(i,j-1). \ [] \end{aligned}$$

Lemma 3 : The post-conditions of the procedure insert are satisfied when it is called and has executed.

Proof

Let the bar $\bar{}$ notation signify generators and groups at the start of the procedure. Note that the groups $G(1), G(2), \ldots, G(j-1)$ and $G(i), G(i+1), \ldots, G(m)$ do not change.

Let $g = [\bar{g}(j), \bar{g}(i)]$ be the argument. We will treat the conditions one at a time.

$\mathbf{SN}(m)$ : The condition $\mathbf{Z}(i-1)$ shows that $\bar{G}(i-1)$ normalizes $\bar{G}(j-1)$. Hence, $g$ normalizes $\bar{G}(j-1)$. Hence, $G(j) = \langle \bar{G}(j-1), g \rangle$ has $G(j-1)$ as a normal subgroup.

For $j < l \leq i-1$, $[g, \bar{g}(l)] \in \bar{G}(l-1)$, by $\mathbf{Z}(i-1)$. Therefore, $\bar{g}(l)$ normalizes $\langle \bar{G}(l-1), g \rangle$. That is, $g(l+1)$ normalizes $G(l)$.

$\mathbf{p}(m)$ : Since $\bar{g}(l)^p \in \bar{G}(l-1)$, $\bar{g}(l)^p \in \langle \bar{G}(l-1), g \rangle$. That is, $g(l)^p \in \bar{G}(l)$, for $l \neq j$. For $l = j$, the pre-condition gives $g(j)^p \in \bar{G}(j-1) = G(j-1)$. The redundant generator $\bar{g}(k)$ is deleted, so each step has index exactly p.

$\mathbf{Z}(i-1)$ : Let $h = \bar{g}(j)^{\bar{g}(i)}$, so that $g = \bar{g}(j)^{-1}h$. By $\bar{\mathbf{Z}}(i-1)$, $\bar{g}(j)$ is central in $\bar{G}(i-1)/G(j-1)$. Therefore $h$ is also central in $\bar{G}(i-1)/G(j-1)$. For $l \geq j$, and $l < r \leq i-1$,

$$[\bar{g}(r), \bar{g}(l)] \in \bar{G}(l-1) \leq \langle \bar{G}(l-1), g \rangle = G(l).$$

It remains to check that $\bar{g}(r)$, $j \leq r \leq i-1$, centralizes g modulo $G(j-1)$.

$$\begin{aligned} [g, \bar{g}(r)] &= g^{-1}\bar{g}(r)^{-1} g \bar{g}(r) \\ &= h^{-1}\bar{g}(j)\bar{g}(r)^{-1}\bar{g}(j)^{-1} h \bar{g}(r) \end{aligned}$$

$= h^{-1}\bar{g}(r)^{-1} h\bar{g}(r)$ modulo $G(j-1)$, by $\mathbf{Z}(i-1)$
$=$ identity modulo $G(j-1)$, since $h$ is central in $\bar{G}(i-1)/G(j-1)$.

$\mathbf{Z}(i,j-1)$ : Since the groups involved have not changed, this condition holds. []

Lemma 4 : The while loop terminates after at most i-j-1 iterations.
Proof

At the $l$-th iteration the while loop considers the commutator $[\bar{g}(j), \bar{g}(i), \bar{g}(i), \ldots, \bar{g}(i)]$, where there are $l$ instances of $\bar{g}(i)$. Since $\frac{G(i)}{G(j-1)}$ has order $p^{i-j+1}$, its nilpotency class is at most i-j. Hence the largest possible value of $l$ without giving the identity modulo $G(j-1)$ is i-j-1. Hence the loop iterates at most i-j-1 times. []

Therefore, the algorithm is correct, and involves at most $O(m^3)$ calls to the procedure insert.

## 3. Preserving a Strong Generating Set

### 3.1 NOTATION

We now need to be aware of the action of the group on a set of points $\Omega$. Elements act on the right, so $\alpha^g$ is the image of $\alpha \in \Omega$ under $g \in G$. The *orbit* of $\alpha$ is $\alpha^G = \{\alpha^g : g \in G\}$ and the *stabilizer* of $\alpha$ is $G_\alpha = \{g \in G : \alpha^g = \alpha\}$. A *base* of the group $G$ is a sequence $[\beta_1, \beta_2, \ldots, \beta_k]$ of points such that only the identity fixes each point of the base. Associated with a base is a chain of stabilizers $G = G^{(1)} \geq G^{(2)} \geq \cdots \geq G^{(k+1)} = identity$ where $G^{(i)} = G_{\beta_1, \beta_2, \ldots, \beta_{i-1}}$. A *strong generating set* of $G$ relative to the base is a set of elements of $G$ that contains a generating set for each stabilizer in the chain.

A base and strong generating set enables us to test membership in a group, and in this case will enable us to determine normed words if the following predicate holds:

**STG**($i$) : $\{ g(1), g(2), \ldots, g(i) \}$ is a strong generating set (relative to the base of $G$) of the group $G(i)$ in the subnormal series.

Let **STG** denote the predicate where **STG**($i$) holds for all values of $i$.

Let fix($g$) be the largest integer such that $g$ fixes $\beta_1, \beta_2, \ldots, \beta_{fix(g)-1}$. Let $\Delta(i,j) = \beta_j^{G(i)^{(j)}}$.

### 3.2 ALGORITHM

The procedure insert is the only part of the previous algorithm that must be altered to preserve a strong generating set. The series of groups produced is the same as the previous version of insert. Hence, the normality, centralizing and $p$-step conditions are still satisfied. Only the choice of generators for the groups is different, in order to retain a strong generating set. Essentially, the element is inserted by bubbling it down into the correct position. Each bubble restores the strong generating property if it is violated. It is only necessary to bubble into positions that occur (non-trivially) in the normed word of the element. We present the modified procedure below, noting only the predicates relevant to the strong generating set.

**Algorithm 2 : Refinement of insert procedure**

**procedure** insert( $g = g(i-1)^{\varepsilon(i-1)} g(i-2)^{\varepsilon(i-2)} \cdots g(1)^{\varepsilon(1)}$, j );
**begin**
{ bar $^-$ notation refers to generators at this point }
{ STG }
let *top* be the highest integer where $\varepsilon(top) \neq 0$;
{by powering $g$ if necessary, we can assume $\varepsilon(top) = 1$}
$k := top$;
**for** $l := top$-1 **downto** $j$ **do**
{STG and $g(k) = \bar{g}(i-1)^{\varepsilon(i-1)} \bar{g}(i-2)^{\varepsilon(i-2)} \cdots \bar{g}(l+1)^{\varepsilon(l+1)}$}
**if** $\varepsilon(l) \neq 0$ **then**
insert_and_preserve( $g(k) g(l)^{\varepsilon(l)}, l$ );
$k := l$;
{ STG }
**if** $k \neq j$ **then**
move $g(k)$ from its present position to between $g(j-1)$ and $g(j)$;
{ STG }
**end;**

**procedure** insert_and_preserve( $g = g(k) g(l)^{\varepsilon(l)}, l$ );
**begin**
{ bar $^-$ notation refers to generators at this point }
delete $g(k)$;
insert $g$ between $g(l-1)$ and $g(l)$;
**if** fix( $\bar{g}(k)$ ) > fix( $\bar{g}(l)$ ) **then**
$g(l+1) := \bar{g}(k)$;
**end;**

## 3.3 PROOF OF CORRECTNESS

We have to prove that the procedures leave the strong generating predicate **STG** invariant. The next two results characterise the elements that extend a strong generating set of $G(i-1)$ to a strong generating set of $G(i)$ in terms of the basic orbits of $G(i-1)$.

Lemma 5 : **STG**($i$) implies $\beta^{g(i)}_{fix(g(i))} \notin \Delta(i-1, fix(g(i)))$
Proof

If $\beta^{g(i)}_{fix(g(i))} \in \Delta(i-1, fix(g(i)))$ then $g(i)$ is redundant in the strong generating set, contrary to $g(i) \notin G(i-1)$. []

Lemma 6 : **STG**($i-1$) $\wedge$ $g$ normalizes $G(i-1) \wedge g^p \in G(i-1) \wedge g \notin G(i-1)$ $\wedge \beta^{g}_{fix(g)} \notin \Delta(i-1, fix(g))$ implies { $g(1), g(2), \ldots, g(i-1), g$ } is a strong generating set of $< G(i-1), g >$

Proof

Since $g$ normalizes $G(i-1)^{(fix(g))}$, $g$ permutes its orbits. Since $g^p \in G(i-1) \wedge g \notin G(i-1)$, $g$ fuses precisely $p$ orbits of $G(i-1)^{(fix(g))}$ into $\Delta(i-1, fix(g))$. So the product of the basic orbits lengths is $p\,|G(i-1)| = |< G(i-1), g >|$. Therefore, $\{ g(1), g(2), \ldots, g(i-1), g \}$ is a strong generating set of $< G(i-1), g >$. []

Next we characterise the level in the stabilizer chain to which an element $g$ belongs. Recall that this level is denoted fix($g$).

Lemma 7 : **STG** implies for $k > l$, $fix(g(k)g(l)) = min(\, fix(g(k)), fix(g(l))\,)$

Proof

Clearly $\text{fix}(\, g(k)g(l)\,) \geq min(\, fix(g(k)), fix(g(l))\,)$.

Suppose that $\text{fix}(\, g(l)\,) < \text{fix}(\, g(k)\,)$. Then $g(k)$ fixes $\beta_{fix(g(l))}$, and $g(l)$ moves it. Therefore $g(k)g(l)$ moves $\beta_{fix(g(l))}$ and $\text{fix}(\, g(k)g(l)\,) = \text{fix}(\, g(l)\,) = \min(\, \text{fix}(g(k)), \text{fix}(g(l))\,)$.

Suppose that $\text{fix}(\, g(l)\,) \geq fix(\, g(k)\,)$. Then $g(k)$ moves $\beta_{fix(g(k))}$ out of $\Delta(k-1, fix(g(k)))$, by lemma 1. Since $l < k$, $\Delta(l, fix(g(k))) \subseteq \Delta(k-1, fix(g(k)))$, so $g(l)$ cannot move $\beta^{g(k)}_{fix(g(k))}$ back into $\Delta(k-1, fix(g(k)))$. Therefore, $g(k)g(l)$ moves $\beta_{fix(g(k))}$ and $fix(g(k)g(l)) = fix(g(k)) = min(\, fix(g(k)), fix(g(l))\,)$. []

Corollary 8 : **STG** $\wedge\, k > l \wedge fix(g(l)) \geq fix(g(k))$ implies $g(k)g(l)$ moves $\beta_{fix(g(k))}$ out of $\Delta(k-1, fix(g(k)))$. []

The next lemma is vital. It tells us what happens with respect to the orbits of the groups when generators are interchanged in the series.

Lemma 9 : If $\beta^g$ is outside $\beta^G$, $\beta^h$ is outside $\beta^{<G,g>}$, $g \notin <G,h>$, $g^p \in G$ and $g$ normalizes $G$, $h^p \in <G,g>$ and $h$ normalizes $<G,g>$, then $\beta^g$ is outside $\beta^{<G,h>}$.

Proof

The orbit $\beta^{<G,g>}$ is the union of $p$ orbits of $G$ each of size $|\beta^G|$ and $\beta^{<G,g,h>}$ is the union of $p$ orbits of $<G,g>$ each of size $|\beta^{<G,g>}|$. Hence $|\beta^{<G,g,h>}| = p^2 |\beta^G|$ and $|<G,g,h>| = p^2 |G|$. Therefore, $G_\beta = <G,g,h>_\beta$.

Suppose that $\beta^g \in \beta^{<G,h>}$. Then there is a power $\hat{h}$ of h such that $\beta^{g\hat{h}} \in \beta^G$. This implies that $g\hat{h} \in G$ because $G_\beta = <G,g,h>_\beta$, contrary to $g \notin <G,h>$. []

The properties of $g(l)^{\varepsilon(l)}$, $0 < \varepsilon(l) < p$, are the same modulo $G(l-1)$ as those of $g(l)$, so the main theorem is as follows.

Theorem 10 : The procedure insert_and_preserve leaves the predicate **STG** invariant.

Proof

Let $g = g(k)g(l)$. Then $g^p \in G(l-1)$, by $\mathbf{Z}(i-1)$. Furthermore, the procedure does not change the groups $G(k)$, $G(k+1)$, ..., $G(m)$, so lemma 3 will imply that **STG** is invariant once we have shown that **STG**(1), **STG**(2), ..., **STG**($k$) are invariant. Furthermore, the relevant generators $g(1)$, $g(2)$, ..., $g(l-1)$ do not change, so **STG**(1), **STG**(2), ..., **STG**($l-1$) are invariant.

There are two cases to consider.

(i) $fix(\bar{g}(k)) \leq fix(\bar{g}(l))$. By $\mathbf{Z}(i-1)$, g normalizes $G(l-1)$. We know that $g^p \in G(l-1)$, $g \notin G(l-1)$ and $\beta^g_{fix(g)}$ is outside of $\Delta(k-1,fix(g))$, by corollary 8, and therefore out of $\Delta(l-1,fix(g))$. Hence, lemma 6 proves that $\{g(1), g(2), \ldots, g(l-1), g\}$ is a strong generating set of $<G(l-1),g>$. Hence, $\bigwedge_{b=1}^{l} \mathbf{STG}(b)$.

Consider r = $l$+1, $l$+2, ..., k. Then $g(r) = \bar{g}(r-1)$. If $\mathrm{fix}(g(r)) \neq fix(g)$ then $\bar{\Delta}(r-2,fix(g(r)))$ is unaltered by the addition of g to $\bar{G}(r-2)$ and so lemma 6 still applies and proves **STG**($r$). If $\mathrm{fix}(g(r)) = \mathrm{fix}(g)$ and $\beta^{g(r)}_{fix(g)}$ is not in the orbit of $\beta_{fix(g)}$ under $<\bar{G}(r-2)^{(fix(g))}, g>$, that is, $\beta^{g(r)}_{fix(g)}$ is outside $\Delta(r-1,fix(g))$, then lemma 6 again proves **STG**($r$).

So we are required to prove that $\beta^{g(r)}_{fix(g)}$ is not in the orbit of $\beta_{fix(g)}$ under $<\bar{G}(r-2)^{(fix(g))}, g>$ when $\mathrm{fix}(g(r)) = \mathrm{fix}(g)$. But this is just lemma 9.

(ii) $fix(\bar{g}(l)) < fix(\bar{g}(k))$.

Thus $g(l+1) = \bar{g}(k)$. (Note that if $g(l+1)$ was $\bar{g}(l)$ then $\mathrm{fix}(g) = \mathrm{fix}(g(l+1))$ and $\beta^g_{fix(g)}$ would be $\beta^{g(l+1)}_{fix(g)}$, which is not outside $\Delta(l,fix(g))$.)

However, $\mathrm{fix}(\bar{g}(k)) \neq fix(g)$ and so $\Delta(l,fix(g(l+1))) = \bar{\Delta}(l-1,fix(\bar{g}(k)))$. Since $\beta^{\bar{g}(k)}_{fix(\bar{g}(k))}$ is outside $\bar{\Delta}(k-1,fix(\bar{g}(k)))$, it is also outside $\Delta(l-1,fix(\bar{g}(k)))$. Hence $\beta^{g(l+1)}_{fix(g(l+1))}$ is outside $\Delta(l-1,fix(g(l+1)))$ and, by lemma 6, **STG**($l$+1).

Also, since $\beta^g_{fix(g)} = \beta^{g(l)}_{fix(g)}$, the orbit $\Delta(l,fix(g))$ equals the orbit $\bar{\Delta}(l,fix(g))$. Hence, for r > $l$+1, if $\mathrm{fix}(g(r)) = \mathrm{fix}(g)$ then $\beta^{g(r)}_{fix(g)}$ is still outside $\Delta(r-1,fix(g))$.

For r > $l$+1, if $\mathrm{fix}(g(r)) = \mathrm{fix}(\bar{g}(k)) = \mathrm{fix}(g(l+1))$ then lemma 9 shows that $\beta^{g(r)}_{fix(\bar{g}(k))}$ is outside $\Delta(r-1,fix(\bar{g}(k)))$. []

Theorem 11 : The procedure insert leaves the predicate **STG** invariant.

Proof

It remains to consider the (possible) last step in the procedure where $g(k)$ is moved between $g(j-1)$ and $g(j)$. But none of the generators skipped by this move are involved in $g(k)$ or the original element g. The same argument as used in theorem 10 for the levels $l$+1, $l$+2, ..., k proves the result. []

Derek Holt and a referee have pointed out a few minor errors in previous drafts. The referee's suggestions have lead to a much neater presentation of section 3. I would like to thank them both.

## References

Butler, G. (1982). Computing in permutation and matrix groups II: backtrack algorithm, *Mathematics of Computation* **39**, 671-680.

Butler, G. (1983). Computing normalizers in permutation groups, *Journal of Algorithms* **4**, 163-175.

Butler, G., Cannon, J.J. (1979). Computing in permutation and matrix groups III: Sylow subgroups, to appear in *Journal of Symbolic Computation*.

Butler, G., Cannon, J.J. (1982). Computing in permutation and matrix groups I: normal closure, commutator subgroup, series, *Mathematics of Computation* **39**, 663-670.

Felsch, V., Neubüser, J. (1979). An algorithm for the computation of conjugacy classes and centralizers in p-groups, in *Symbolic and Algebraic Manipulation (Marseille, 1979)*. Edited by Edward W. Ng. Lecture Notes in Computer Science, volume 72, Springer-Verlag, Berlin, pp.452-465.

Hoffman, C.M. (1982). *Group-Theoretical Algorithms and Graph Isomorphism*, Lecture Notes in Computer Science, volume 136, Springer-Verlag, Berlin.

Holt, D.F. (1984). The calculation of the Schur multiplier of a permutation group, in *Computational Group Theory* (Proceedings of LMS Symposium on Computational Group Theory, Durham, 1982). Edited by Michael Atkinson. Academic Press, New York, pp.307-319.

Laue, R., Neubüser, J., Schoenwaelder, U. (1984). Algorithms for finite soluble groups and the SOGOS system, in *Computational Group Theory* (Proceedings of LMS Symposium on Computational Group Theory, Durham, 1982). Edited by Michael Atkinson. Academic Press, New York, pp.105-135.

Leon, J.S. (1980). On an algorithm for finding a base and strong generating set for a group given by generating permutations, *Mathematics of Computation* **35**, 941-974.

Macdonald, I.D. (1968). *The Theory of Groups*, Oxford University Press, Oxford.

Wielandt, H. (1964). *Finite Permutation Groups*, Academic Press, New York.